COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco
University of Massachusetts Amherst. Spring 2020.
Lecture 16
Last Class: Low-Rank Approximation, Eigendecomposition, and PCA

- Can approximate data lying close to in a $k$-dimensional subspace by projecting data points into that space.
- Finding the best $k$-dimensional subspace via eigendecomposition (PCA).
- Measuring error in terms of the eigenvalue spectrum.

This Class: Finish Low-Rank Approximation and Connection to the singular value decomposition (SVD)

- Finish up PCA – runtime considerations and picking $k$.
- View of optimal low-rank approximation using the SVD.
- Applications of low-rank approximation beyond compression.
Set Up: Assume that data points \( \tilde{x}_1, \ldots, \tilde{x}_n \) lie close to any \( k \)-dimensional subspace \( \mathcal{V} \) of \( \mathbb{R}^d \). Let \( X \in \mathbb{R}^{n \times d} \) be the data matrix.

Let \( \vec{v}_1, \ldots, \vec{v}_k \) be an orthonormal basis for \( \mathcal{V} \) and \( V \in \mathbb{R}^{d \times k} \) be the matrix with these vectors as its columns.

- \( VV^T \in \mathbb{R}^{d \times d} \) is the projection matrix onto \( \mathcal{V} \).
- \( X \approx X(VV^T) \). Gives the closest approximation to \( X \) with rows in \( \mathcal{V} \).

\( \tilde{x}_1, \ldots, \tilde{x}_n \in \mathbb{R}^d \): data points, \( X \in \mathbb{R}^{n \times d} \): data matrix, \( \vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d \): orthogonal basis for subspace \( \mathcal{V} \), \( V \in \mathbb{R}^{d \times k} \): matrix with columns \( \vec{v}_1, \ldots, \vec{v}_k \).
V minimizing \(\|X - XVV^T\|_F^2\) is given by:

\[
\arg\max_{\text{orthonormal } V \in \mathbb{R}^{d \times k}} \|XV\|_F^2 = \sum_{j=1}^{k} \|X\vec{v}_j\|_2^2
\]

Solution via eigendecomposition: Letting \(V_k\) have columns \(\vec{v}_1, \ldots, \vec{v}_k\) corresponding to the top \(k\) eigenvectors of the covariance matrix \(X^TX\),

\[
V_k = \arg\max_{\text{orthonormal } V \in \mathbb{R}^{d \times k}} \|XV\|_F^2
\]

- Proof via Courant-Fischer and greedy maximization.
- Approximation error is \(\|X\|_F^2 - \|XV_k\|_F^2 = \sum_{i=k+1}^{d} \lambda_i(X^TX)\).

\(\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d\): data points, \(X \in \mathbb{R}^{n \times d}\): data matrix, \(\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d\): orthogonal basis for subspace \(\mathcal{V}\). \(V \in \mathbb{R}^{d \times k}\): matrix with columns \(\vec{v}_1, \ldots, \vec{v}_k\).
LOW-RANK APPROXIMATION VIA EIGENDECOMPOSITION

\[ X^T X = \begin{bmatrix} \tilde{v}_1 & \tilde{v}_2 \end{bmatrix} V^T \Lambda V \]

\[ \begin{bmatrix} \lambda_1 & \lambda_2 & \ldots & \lambda_d \end{bmatrix} \]

\[ V^T \]

\[ \text{d-dimensional space} \]

\[ \text{k-dim. subspace } \mathcal{V} \]
SPECTRUM ANALYSIS

Plotting the spectrum of the covariance matrix $X^T X$ (its eigenvalues) shows how compressible $X$ is using low-rank approximation (i.e., how close $\vec{x}_1, \ldots, \vec{x}_n$ are to a low-dimensional subspace).

- Choose $k$ to balance accuracy and compression.
- Often at an ‘elbow’.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $X \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $X^T X$, $V \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$. 

\[ X^T X = \vec{v}_1 \vec{v}_2 \ldots \vec{v}_k V \]
Exercise: Show that the eigenvalues of $X^TX$ are always positive.

Hint: Use that $\lambda_j = \overline{v}_j^T X^T X \overline{v}_j$. 

784 dimensional vectors
**Recall:** Low-rank approximation is possible when our data features are correlated.

Our compressed dataset is $\mathbf{C} = \mathbf{XV}_k$ where the columns of $\mathbf{V}_k$ are the top $k$ eigenvectors of $\mathbf{X}^T\mathbf{X}$.

**What is the covariance of $\mathbf{C}$?**

$$\mathbf{C}^T \mathbf{C} = \mathbf{V}_k^T \mathbf{X}^T \mathbf{X} \mathbf{V}_k = \mathbf{V}_k^T \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^T \mathbf{V}_k = \boldsymbol{\Lambda}_k$$

**Covariance becomes diagonal.** I.e., all correlations have been removed. Maximal compression.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$. 
What is the runtime to compute an optimal low-rank approximation?

- Computing the covariance matrix $\mathbf{X}^T\mathbf{X}$ requires $O(nd^2)$ time.
- Computing its full eigendecomposition to obtain $\mathbf{v}_1, \ldots, \mathbf{v}_k$ requires $O(d^3)$ time (similar to the inverse $(\mathbf{X}^T\mathbf{X})^{-1}$).

Many faster iterative and randomized methods. Runtime is roughly $\tilde{O}(ndk)$ to output just to top $k$ eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$.

- Will see in a few classes (power method, Krylov methods).

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\mathbf{v}_1, \ldots, \mathbf{v}_k$. 
The Singular Value Decomposition (SVD) generalizes the eigendecomposition to asymmetric (even rectangular) matrices. Any matrix $X \in \mathbb{R}^{n \times d}$ with rank$(X) = r$ can be written as $X = U\Sigma V^T$.

- $U$ has orthonormal columns $\vec{u}_1, \ldots, \vec{u}_r \in \mathbb{R}^n$ (left singular vectors).
- $V$ has orthonormal columns $\vec{v}_1, \ldots, \vec{v}_r \in \mathbb{R}^d$ (right singular vectors).
- $\Sigma$ is diagonal with elements $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ (singular values).

The ‘swiss army knife’ of modern linear algebra.
Writing $X \in \mathbb{R}^{n \times d}$ in its singular value decomposition $X = U \Sigma V^T$:

$$X^T X = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T \quad \text{(the eigendecomposition)}$$

Similarly: $XX^T = U \Sigma V^T V \Sigma U^T = U \Sigma^2 U^T$.

The left and right singular vectors are the eigenvectors of the covariance matrix $X^T X$ and the gram matrix $XX^T$ respectively.

So, letting $V_k \in \mathbb{R}^{d \times k}$ have columns equal to $\vec{v}_1, \ldots, \vec{v}_k$, we know that $XV_k V_k^T$ is the best rank-$k$ approximation to $X$ (given by PCA).

What about $U_k U_k^T X$ where $U_k \in \mathbb{R}^{n \times k}$ has columns equal to $\vec{u}_1, \ldots, \vec{u}_k$?

Gives exactly the same approximation!

---

**X** \in \mathbb{R}^{n \times d}: data matrix, **U** \in \mathbb{R}^{n \times \text{rank}(X)}: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), **V** \in \mathbb{R}^{d \times \text{rank}(X)}: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), **Σ** \in \mathbb{R}^{\text{rank}(X) \times \text{rank}(X)}: positive diagonal matrix containing singular values of **X**.
The best low-rank approximation to $X$:

$$X_k = \arg \min_{\text{rank } - k} \| X - B \|_F$$

is given by:

$$X_k = XV_k V_k^T = U_k U_k^T X = U_k \Sigma_k V_k^T$$

Correspond to projecting the rows (data points) onto the span of $V_k$ or the columns (features) onto the span of $U_k$
The best low-rank approximation to $X$:

$$X_k = \arg \min_{B \in \mathbb{R}^{n \times d}} \|X - B\|_F$$

is given by:

$$X_k = XV_kV_k^T = U_kU_k^TX = U_k\Sigma_kV_k^T$$

**X** $\in \mathbb{R}^{n \times d}$: data matrix, **U** $\in \mathbb{R}^{n \times \text{rank}(X)}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), **V** $\in \mathbb{R}^{d \times \text{rank}(X)}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), **Σ** $\in \mathbb{R}^{\text{rank}(X) \times \text{rank}(X)}$: positive diagonal matrix containing singular values of **X**.
THE SVD AND OPTIMAL LOW-RANK APPROXIMATION

\[ X \in \mathbb{R}^{n \times d} \]: data matrix, \( U \in \mathbb{R}^{n \times \text{rank}(X)} \): matrix with orthonormal columns \( \vec{u}_1, \vec{u}_2, \ldots \) (left singular vectors), \( V \in \mathbb{R}^{d \times \text{rank}(X)} \): matrix with orthonormal columns \( \vec{v}_1, \vec{v}_2, \ldots \) (right singular vectors), \( \Sigma \in \mathbb{R}^{\text{rank}(X) \times \text{rank}(X)} \): positive diagonal matrix containing singular values of \( X \).
Rest of Class: Examples of how low-rank approximation is applied in a variety of data science applications.

• Used for many reasons other than dimensionality reduction/data compression.
Consider a matrix $X \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-$k$ (i.e., well approximated by a rank $k$ matrix). Classic example: the Netflix prize problem.

**Solve:** $Y = \arg \min_{\text{rank } - k B} \sum_{\text{observed } (j,k)} [X_{j,k} - B_{j,k}]^2$

Under certain assumptions, can show that $Y$ well approximates $X$ on both the observed and (most importantly) unobserved entries.
Dimensionality reduction embeds $d$-dimensional vectors into $d'$ dimensions. But what about when you want to embed objects other than vectors?

- Documents (for topic-based search and classification)
- Words (to identify synonyms, translations, etc.)
- Nodes in a social network

**Usual Approach:** Convert each item into a high-dimensional feature vector and then apply low-rank approximation.
EXAMPLE: LATENT SEMANTIC ANALYSIS

Term Document Matrix $X$

Low-Rank Approximation via SVD

$X \approx U_k \Sigma_k V_k^T$
If the error $\|X - YZ^T\|_F$ is small, then on average,

$$X_{i,a} \approx (YZ^T)_{i,a} = \langle \vec{y}_i, \vec{z}_a \rangle.$$ 

- I.e., $\langle \vec{y}_i, \vec{z}_a \rangle \approx 1$ when $doc_i$ contains word $a$.
- If $doc_i$ and $doc_j$ both contain word $a$, $\langle \vec{y}_i, \vec{z}_a \rangle \approx \langle \vec{y}_j, \vec{z}_a \rangle = 1$. 

\[\text{Term Document Matrix } X\]

\[\begin{array}{cccc}
doc_1 & yes & no & yes & no \\
doc_2 & no & yes & yes & no \\
... & ... & ... & ... & ... \\
doc_n & yes & no & no & yes \\
\end{array}\]

\[\text{Low-Rank Approximation via SVD}\]

$X \approx YZ^T$
If $doc_i$ and $doc_j$ both contain $word_a$, $\langle \vec{y}_i, \vec{z}_a \rangle \approx \langle \vec{y}_j, \vec{z}_a \rangle = 1$

Another View: Each column of $Y$ represents a ‘topic’. $\vec{y}_i(j)$ indicates how much $doc_i$ belongs to topic $j$. $\vec{z}_a(j)$ indicates how much $word_a$ associates with that topic.
Just like with documents, $\vec{z}_a$ and $\vec{z}_b$ will tend to have high dot product if $\text{word}_i$ and $\text{word}_j$ appear in many of the same documents.

In an SVD decomposition we set $Z = \Sigma_k V_k^T$.

The columns of $V_k$ are equivalently: the top $k$ eigenvectors of $X^T X$. The eigendecomposition of $X^T X$ is $X^T X = V \Sigma^2 V^T$.

What is the best rank-$k$ approximation of $X^T X$? I.e.

$\arg \min_{\text{rank} - k \ B} \| X^T X - B \|_F$

$X^T X = V_k \Sigma_k^2 V_k^T = ZZ^T$. 
LSA gives a way of embedding words into $k$-dimensional space.

- Embedding is via low-rank approximation of $X^TX$: where $(X^TX)_{a,b}$ is the number of documents that both $\text{word}_a$ and $\text{word}_b$ appear in.

- Think about $X^TX$ as a similarity matrix (gram matrix, kernel matrix) with entry $(a, b)$ being the similarity between $\text{word}_a$ and $\text{word}_b$.

- Many ways to measure similarity: number of sentences both occur in, number of times both appear in the same window of $w$ words, in similar positions of documents in different languages, etc.

- Replacing $X^TX$ with these different metrics (sometimes appropriately transformed) leads to popular word embedding algorithms: word2vec, GloVe, fastTest, etc.
Note: word2vec is typically described as a neural-network method, but it is really just low-rank approximation of a specific similarity matrix. *Neural word embedding as implicit matrix factorization*, Levy and Goldberg.
Summary:

- Can use the SVD to understand optimal low-rank approximation in terms of the dual row/column projection view: $XV_kV_k^T = U_kU_k^TX = U_k\Sigma_kV_k^T$.
- A generalization of eigendecomposition: singular vectors are eigenvectors of $XX^T$ and $X^TX$.
- Applications to low-rank approximation to matrix completion and entity embeddings.